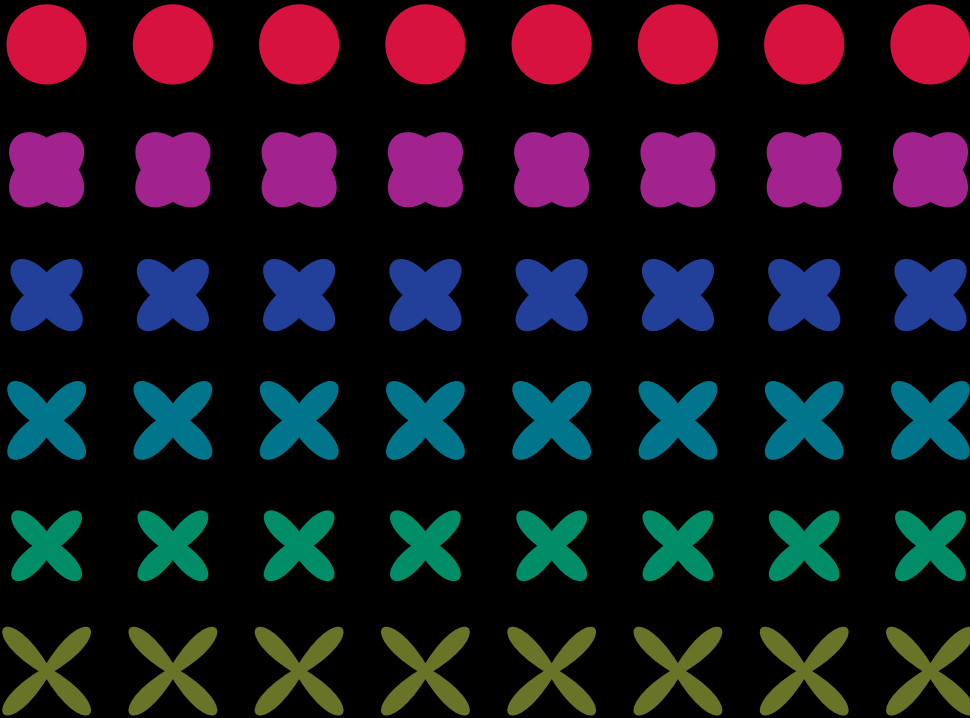


IUNA

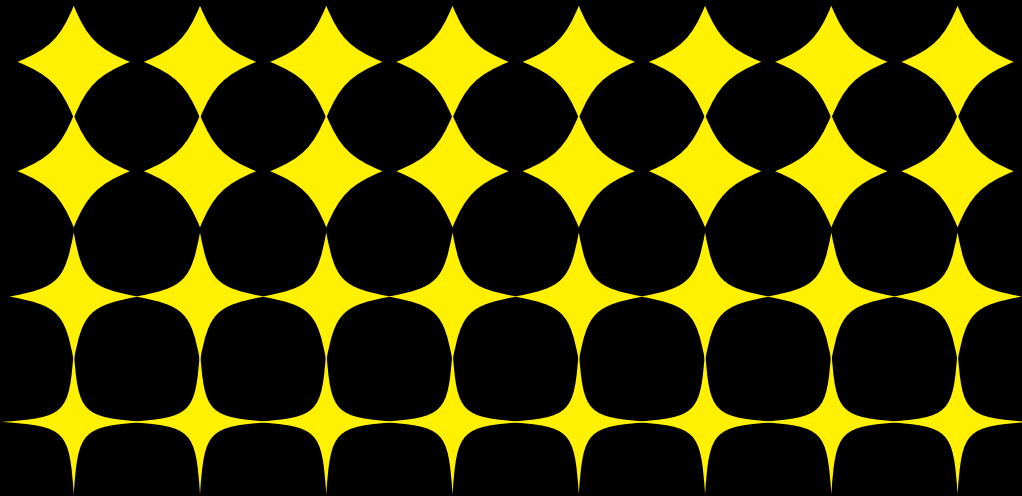
Instituto Universitario
Nacional del Arte

RIM

revista de investigación multimedia
año 3. número 3. marzo de 2011



3



IUNA

Instituto Universitario Nacional del Arte

editorial

... cabe tener en cuenta las posibilidades que pueden llegar a seguirse de los encuentros entre ciencia y las prácticas artísticas. En muchos casos, los artistas toman inspiración en los hallazgos de la ciencia, o investigan de manera creativa en algunos campos tecno-científicos (por ejemplo, y en estos momentos, la ingeniería genética, la nanotecnología, la matemática del límite, el software avanzado, la astronomía, la robótica, la inteligencia artificial, son campos en los que unos u otros artistas han encontrado inspiración directa y a partir de los que de hecho realizan alguna investigación creativa suficientemente relevante, cuando menos desde el punto de vista artístico).

(José Luis Brea, cultura_RAM, 2007)

En el año del Bicentenario (1810-2010) hacemos entrega del tercer número de la Revista de Investigación Multimedia del Área Transdepartamental de Artes Multimediales del Instituto Universitario Nacional del Arte.

La misma se presenta como una edición especial de los documentos de avance de la primera etapa del proyecto PICTO “Diseño y desarrollo de aplicaciones e interfases de Realidad Aumentada destinadas a síntesis y procesamiento de audio digital”, aprobado por la Agencia Nacional de Promoción Científica y Tecnológica - Fondo para la Investigación Científica y Tecnológica (Ministerio de Ciencia, Tecnología e Innovación Tecnológica - Presidencia de la Nación).

El proyecto parte del paradigma “continuo realidad-virtualidad” (Milgram, Kishino y Takemura, 1994) el cual ubica a los entornos virtuales en diversas categorías, cuyo rango se extiende desde la realidad física hasta la realidad virtual. Situados en este paradigma, entendemos por entornos de realidad aumentada a aquellos que logran conjugar elementos virtuales con la realidad física que nos rodea. Dentro de esta clasificación se encuadra la mayor parte de las experiencias de multimedia escénica, como las instalaciones, las *performances* y las intervenciones interactivas en las cuales realidad y virtualidad se funden.

En esta entrega volcamos los resultados de siete artículos que avanzan sobre los tópicos planteados. “Nuevas caracterizaciones de la actividad musical en el aula” de Prof. Carmelo Saitta abre un espacio de análisis sobre la vinculación música - espacio áulico. “Composición asistida en entorno PD” de los Dres. Pablo Cetta y Oscar Pablo Di Liscia, y “Medidas de similitud entre sucesiones ordenadas de grados cromáticos” del Dr. Oscar Pablo Di Liscia presentan técnicas de composición y organización del material musical en el entorno de composición en tiempo real *Pure data*. “Desarrollo de un sistema óptico para interfaces tangibles (mesa con pantalla reactiva)” y “Diseño de interface para el desarrollo de una pantalla sensible al tacto con aplicación musical” del Ing. Emiliano Causa exponen, respectivamente, el desarrollo de una mesa con pantalla sensible al tacto y el diseño de una interface de pantalla sensible al tacto (del tipo multitacto) aplicada a un editor gestual de música. “Técnicas de síntesis y procesamiento de sonido y su aplicación en tiempo real” del Lic. Matías Romero Costas repasa algunas de las técnicas de síntesis y procesamiento de sonido más utilizadas y difundidas, desde un punto de vista teórico, a través de ejemplos de aplicación, implementados en el entorno de programación Max-MSP. Por último, el artículo “Técnicas de programación vinculadas a la realidad aumentada y a las interfaces tangibles” del DCV Tarcisio Lucas Pirotta, presenta las principales funciones y bloques de programación disponibles en los paquetes de librería ARToolKit y reacTIVision, para su aplicación a proyectos de realidad aumentada e interfaces tangibles.

Edición especial dedicada al proyecto PICTO “Diseño y desarrollo de aplicaciones e interfases de Realidad Aumentada destinadas a síntesis y procesamiento de audio digital”, en el año del Bicentenario (1810-2010).

Comité Editorial de RIM

staff



Instituto Universitario Nacional del Arte

Rectora

Prof. Liliana Beatriz Demaio

Vicerrectora

Prof. Susana Pires Mateu

Secretaría General

Prof. María Martha Gigena

Secretaría de Asuntos Jurídico-Legales

Dra. Clara María Picasso Achaval

Secretaría de Asuntos Académicos

Prof. Oscar Steimberg

Secretaría de Investigación y Posgrado

Asesora Graciana Vázquez Villanueva

Secretaría de Extensión y Bienestar Estudiantil

Mg. Fernando Lerman

Secretaría de Desarrollo y Vinculación Institucional

Arq. Daniel Wolkowicz

Secretaría de Asuntos Económico-Financieros

Cont. Eduardo Jorge Auzmendi

Secretaría de Asuntos Administrativos

Asesor Federico Hernán Tessore

Secretaría de Infraestructura y Planeamiento

Arq. Nicolás Escobari



Área Transdepartamental de Artes Multimediales

Director

Prof. Carmelo Saitta

Secretario Académico

Dr. Pablo Cetta

Secretario Administrativo

Dr. Roberto Abait

Coordinación de Actividades de Investigación y Posgrado

Mg. Raúl Lacabanne

Coordinación de Actividades de Extensión y Bienestar Estudiantil

Prof. Gumersindo Jerónimo de Jesús Serrano Gómez

RIM

Director

Prof. Carmelo Saitta

Secretario de redacción

Dr. Pablo Cetta

Comité editorial

Ing. Emiliano Causa

Dr. Pablo Cetta

Dr. Pablo Di Liscia

Mg. Raúl Lacabanne

Prof. Gumersindo Jerónimo de Jesús Serrano Gómez

Arq. Daniel Wolkowicz

Colaboran en este número

Carmelo Saitta

Pablo Cetta

Pablo Di Liscia

Emiliano Causa

Matías Romero Costas

Tarcisio Lucas Pirotta

Diseño

Arq. Daniel Wolkowicz

Corrección de textos

Rossana Cabrera

RIM es una publicación del Área

Transdepartamental de Artes Multimediales
del IUNA

Yatay 843, Ciudad Autónoma de Buenos Aires,
República Argentina

Todos los derechos reservados

ISSN 1850-2954

Impreso en New Press Grupo Impresor S.A.

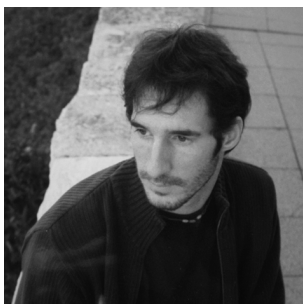
Paraguay 278, Avellaneda

Provincia de Buenos Aires

Marzo de 2011

Índice

Nuevas caracterizaciones de la actividad musical en el aula Carmelo Saitta	5
Composición asistida en entorno PD Dr. Pablo Cetta y Dr. Oscar Pablo Di Liscia	15
Medidas de similitud entre sucesiones ordenadas de grados cromáticos Dr. Oscar Pablo Di Liscia	31
Desarrollo de un sistema óptico para interfaces tangibles (mesa con pantalla reactiva) Ing. Emiliano Causa	45
Diseño de interface para el desarrollo de una pantalla sensible al tacto con aplicación musical Ing. Emiliano Causa	54
Técnicas de síntesis y procesamiento de sonido y su aplicación en tiempo real Lic. Matías Romero Costas	69
Técnicas de programación vinculadas a la realidad aumentada y a las interfaces tangibles DCV Tarcisio Lucas Pirotta	85



MATÍAS ROMERO COSTAS

Compositor, artista multimedia, docente e investigador. Licenciado en Composición; profesor de Armonía, Contrapunto y Morfología Musical; y profesor en Producción Multimedial, diplomado en la Universidad Nacional de La Plata.

Trabaja como docente de grado, posgrado e investigador en la UNLP, el IUNA y ENERC (INCAA).

Forma parte del grupo “Proyecto Biopus” con el que ha presentado obras de net-art, video arte, *performances* e instalaciones interactivas. Sus obras han sido presentadas en Argentina y el exterior. Ha recibido premios nacionales e internacionales.

Técnicas de síntesis y procesamiento de sonido y su aplicación en tiempo real

Matías Romero Costas

Palabras claves

Audio digital, síntesis de sonido, procesamiento, tiempo real

Resumen

En el presente artículo estudiaremos algunas de las técnicas de síntesis y procesamiento de sonido más utilizadas y difundidas, desde un punto de vista teórico y a través de ejemplos de aplicación, implementados en el entorno de programación Max-MSP. Este software fue creado originalmente en el IRCAM en 1996, y desarrollado actualmente por Cycling74, es un entorno de programación pensado como una herramienta para la composición algorítmica. La elección de este entorno se debe a que es una herramienta muy poderosa para síntesis, generación, procesamiento y ejecución de sonido, música e imagen en tiempo real. Por otro lado, su lenguaje de programación con objetos gráficos lo convierte en una herramienta muy didáctica para abordar estos temas.

Finalmente analizaremos la programación de una instalación sonora interactiva, donde se combinan varios de estos módulos de generación y procesamiento, para crear un instrumento complejo.

1. Síntesis aditiva

La síntesis aditiva es una de las primeras técnicas de síntesis, y tiene sus orígenes en la música electrónica.

Esta técnica encuentra sus fundamentos en los descubrimientos del físico francés Jean Baptiste Joseph Fourier (1768-1830), quien reveló que cualquier onda compleja periódica puede ser descompuesta en una serie de ondas simples o sinusoidales. Basada en estos principios, la síntesis aditiva consiste en construir sonidos complejos a partir de la sumatoria de sonidos sinusoidales de diferentes amplitudes, frecuencias y fases.

La síntesis aditiva, expresada en forma de ecuación, queda definida como:

$$y(n) = \sum_{n=0}^N A_n \cos(\omega_n + \theta_n)$$

Donde n es el número de muestra, A_n es la amplitud de la muestra n , ω es su frecuencia angular y θ su fase inicial.

Otra manera de entenderlo es a través de un diagrama de flujo:

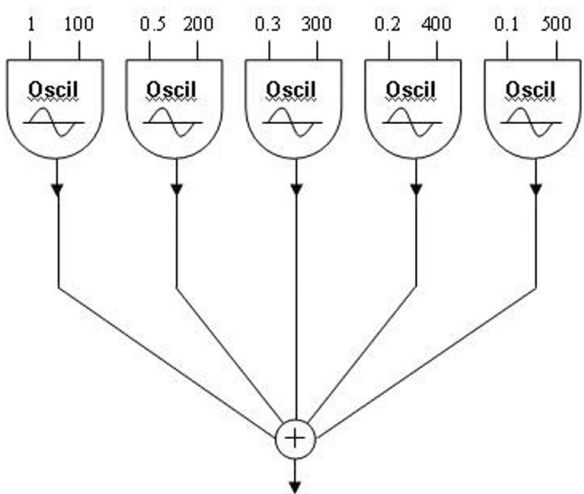


Figura 1: esquema o gráfico de flujo de un instrumento de síntesis aditiva

1.1. Osciladores

El elemento fundamental utilizado en la síntesis aditiva, al igual que en muchas otras técnicas de síntesis, es el oscilador.

Un oscilador digital es una unidad generadora que lee cíclicamente una función para generar una onda periódica. Existen dos formas de implementar un oscilador, una es especificar la forma de onda como una función matemática dependiente del tiempo. Para ello, el algoritmo tiene que calcular el valor de la función matemática para cada una de las muestras. Por ejemplo, si queremos generar una onda sinusoidal, la función que debe calcularse para cada una de las muestras es la siguiente:

$$y(n) = A \sin(2\pi f \frac{n}{sr} + \theta)$$

En donde: n es el número de muestra; A es la amplitud; f la frecuencia (Hz); sr es la frecuencia de muestreo (*sampling rate*) y θ es la fase.

Otro método utilizado, más veloz y con menor consumo de procesamiento, es el llamado oscilador de acceso a tablas (*wavetable lookup oscillator*), que consiste en almacenar en memoria un solo ciclo de forma de onda cualquiera. En ese bloque, o tabla, se almacenan una secuencia de números, cada una de las cuales corresponde al valor de amplitud de cada muestra de la forma de onda. El oscilador, por tanto, solo debe leer de esta tabla los valores sucesivos para generar la forma de onda correspondiente. Una vez que se lee la última muestra, el oscilador vuelve a comenzar desde el principio.

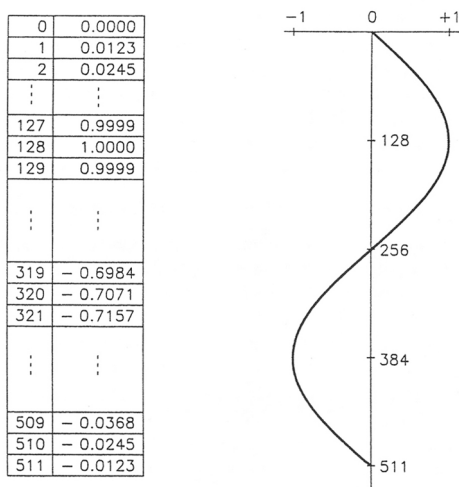


Figura 2: tabla de onda (izquierda) de una senoide (derecha)

En la Figura 2, observamos una tabla de onda (*wavetable*) cuyo tamaño es de 512 muestras. La primera columna indica los índices de cada muestra y la segunda los valores de amplitud asociados a ellas. Para obtener los valores de amplitud que representan a la onda sinusoidal, el oscilador vinculado a la tabla accede, mediante un puntero, a cada uno de los índices de la tabla. En este caso los valores de amplitud están “normalizados”, es decir que están ajustados en el rango que va entre 0 y 1. Pero también, al ser esta una onda sinusoidal bipolar, su hemiciclo negativo se encuentra entre 0 y -1. Es decir, que el oscilador que lea esta tabla devolverá a una velocidad igual a la frecuencia de muestreo valores de amplitud entre 1 y -1 que representan un ciclo de una función senoidal.

En el siguiente esquema vemos la representación de un oscilador en el que se muestra la unidad generadora con un símbolo en su interior que designa la forma de onda. El valor que ingresa por la izquierda representa la amplitud pico de la señal; el valor de la derecha, la frecuencia a la que se repite el ciclo de la onda. Otro parámetro posible a determinar es la fase, que establece en qué punto de la onda el oscilador comienza a leer las muestras.



Figura 3: esquema de un oscilador. En este caso una senoide

1.2. Implementación de la síntesis aditiva en tiempo real

En el gráfico siguiente observamos la programación de una síntesis aditiva con cinco osciladores, y a la derecha la representación gráfica de su espectro, su forma de onda y su sonograma.

En la parte superior, los objetos de mensajes especifican la frecuencia fundamental del sonido resultante (100 Hz, 350 Hz y 1500 Hz). Esta frecuencia es luego multiplicada por 1, 2, 3, 4 y 5 respectivamente para establecer la frecuencia de los parciales de cada uno de los cinco osciladores, en este caso en relación armónica (múltiplos enteros con respecto a la fundamental).

Si quisiéramos obtener un espectro inarmónico bastaría modificar los valores de la multiplicación por números reales.

Antes de sumarse en la salida, los armónicos son multiplicados por un factor de 0.2 para evitar la saturación.

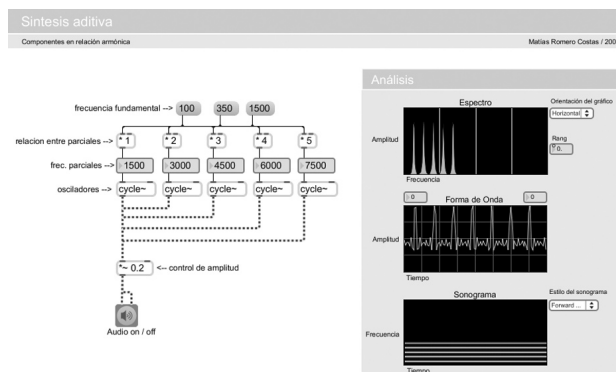


Figura 4

El ejemplo que sigue incluye el control de la amplitud de cada uno de los componentes del espectro a partir de una envolvente dinámica, cuyo perfil puede editarse de manera muy simple con el mouse.

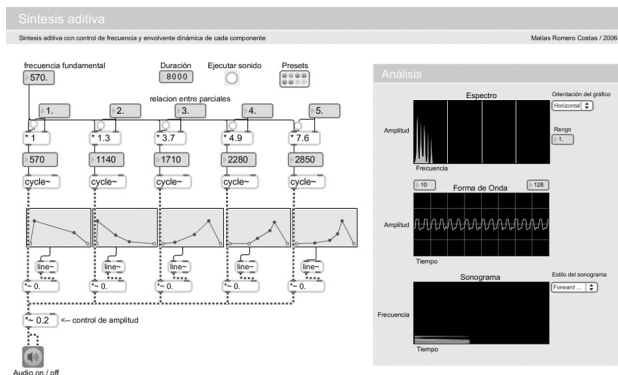


Figura 5

Si analizamos cualquier sonido de la naturaleza, o los que provienen de los instrumentos musicales, vamos a encontrar un espectro sumamente complejo, con cientos o miles de componentes, donde cada uno de ellos evoluciona de manera diferente a lo largo del tiempo. El ejemplo anterior solo intenta demostrar el funcionamiento básico de la síntesis aditiva, y para conseguir un resultado sonoro realmente interesante, se debería utilizar una cantidad mucho mayor de osciladores.

2. Modulación de amplitud

La variación periódica o cuasi periódica de la amplitud de una señal, denominada portadora, en función de la variación de otra, la moduladora, es conocida como modulación de amplitud.

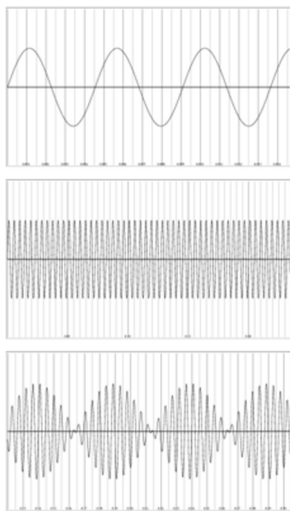


Figura 6: de arriba hacia abajo, señal moduladora, señal portadora y resultado de la modulación.

En audio digital, la manera de implementar la modulación es multiplicando las dos formas de onda, la portadora y la moduladora. Como vimos en el caso de la síntesis aditiva, cuando dos o más formas de onda se suman, lo mismo sucede con sus espectros. El espectro resultante se

conforma con la suma de cada una de las frecuencias de los componentes simples. Sin embargo, cuando multiplicamos dos formas de onda el resultado en el espectro no se corresponde con la multiplicación de ambos, sino con su convolución.

Para poder entenderlo tomemos por caso la multiplicación de dos sinusoides de frecuencias f_1 y f_2 . El espectro resultante estará conformado por dos componentes iguales a:

$$f_1 + f_2$$

$$f_1 - f_2$$

En la Figura 7 aparecen, en la parte superior, los espectros de las señales a ser convolucionadas, cuyas frecuencias son, respectivamente $f_1 = 1000$ Hz y $f_2 = 100$ Hz. El resultado de la convolución, en la parte inferior, nos muestra un espectro complejo, con dos parciales de 900 y 1100 Hz, ubicados a los lados de f_1 , y a una distancia igual a f_2 , a la derecha e izquierda de esta.

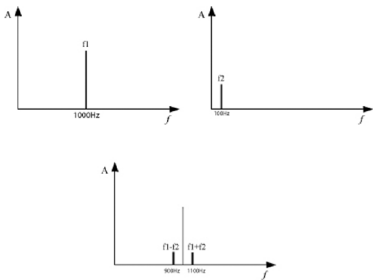


Figura 7: convolución de dos espectros simples.

En el caso de que la moduladora sea de baja frecuencia, por ejemplo 1 Hz, lo que escuchamos no es un espectro complejo sino una sola componente, cuya frecuencia es el promedio de las dos anteriores, con un batido de 1Hz. Este fenómeno solo sucede en el nivel perceptivo y está relacionado con los umbrales de percepción de la frecuencia.

2.1. Trémolo

Se utiliza este término en música para designar el modo de ejecución de algunos instrumentos acústicos, y consiste en alterar la amplitud del sonido repetidamente a gran velocidad. La velocidad de la modulación, al ser controlada por un LFO² (velocidad de control), está en el rango de entre 0.5 seg. y 1 seg. Esto quiere decir que lo que se ve afectado al nivel perceptivo es la cualidad de superficie del sonido.

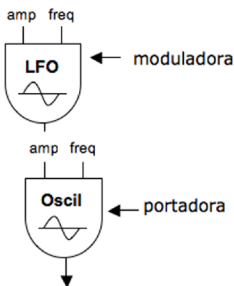


Figura 8: diagrama de flujo del trémolo

En un instrumento, los parámetros que suelen manejarse en la modulación de amplitud son:

- La velocidad de la modulación, controlada por la frecuencia de la moduladora.
- La profundidad de la modulación, que se corresponde con la amplitud de la moduladora.
- La silueta, determinada por la forma de onda de la moduladora.

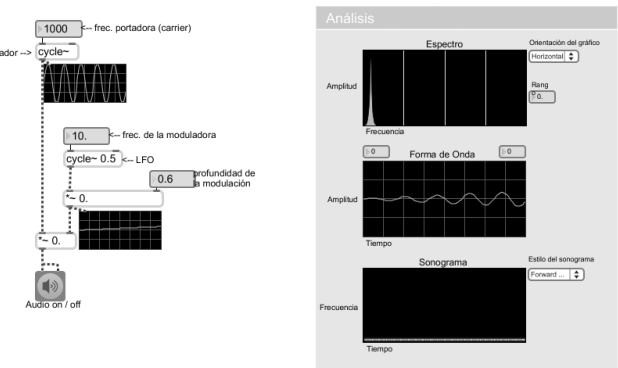


Figura 9: implementación en Max MSP de un trémolo

Si observamos con detenimiento el ejemplo anterior, podemos notar que la profundidad de modulación no está del todo bien resuelta, veamos porqué. Si la amplitud máxima de la portadora es igual a 1, y la profundidad de la modulación es igual a 0.5, los valores de amplitud máximos que resulten de la modulación estarán en el rango de 0 a 0.5. Pero, si queremos que la profundidad sea mayor, sin afectar la amplitud máxima de la portadora, es necesario invertir esta relación. De esta manera nos aseguramos que la modulación “cale” más o menos profundo en la silueta de la señal portadora. Esto queda resuelto en el ejemplo que sigue.

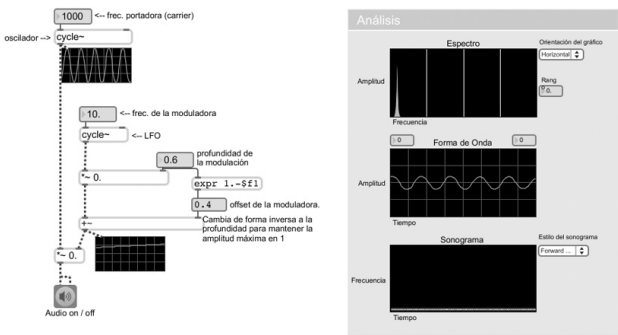


Figura 10: profundidad de modulación normalizada

2.2. Modulación en anillo

Cuando la modulación de amplitud se genera utilizando una moduladora cuya frecuencia se encuentra en rango audible (mayor a 20 Hz) se lo denomina modulación en anillo. Como puede deducirse, la envolvente afecta una porción temporal de la señal

portadora que está en el orden de entre los 0.5 segundos y los 0.0005 segundos.

La aplicación que veremos a continuación es una modulación en anillo aplicada sobre un archivo de audio levantado del disco rígido y ejecutado por el objeto `splay~`.

Sabemos que la multiplicación de dos formas de onda es equivalente a la convolución de sus espectros, y hemos visto que en el caso de multiplicar dos sinusoides con frecuencias f_1 y f_2 , obtenemos un espectro con dos parciales iguales a $f_1 - f_2$ y $f_1 + f_2$. En el caso de multiplicar un sonido complejo con una senoide, la operación de suma y resta se produce por cada uno de los componentes de la portadora. De esta forma se duplica la cantidad de parciales, y dependiendo de la frecuencia de la moduladora, esta técnica permite convertir muy fácilmente un sonido armónico en inarmónico.

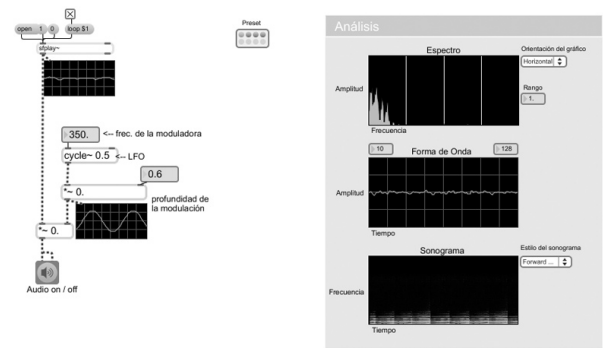


Figura 11: modulación en anillo sobre un archivo de audio

Si trabajamos con señales periódicas, existen ciertas relaciones entre la frecuencia fundamental de la portadora y la frecuencia de la moduladora que tienen resultados muy interesantes. Por ejemplo, si la frecuencia de la moduladora es la mitad de la frecuencia fundamental de la portadora, el espectro resultante solo va a contener armónicos impares.

Otro caso especial se produce cuando la frecuencia de la moduladora es múltiplo de la fundamental de la portadora. Allí el resultado es una transposición de la envolvente espectral.

3. Síntesis por FM

Es posible utilizar una señal moduladora para crear desviaciones periódicas de frecuencia sobre otro oscilador. Si el modulador se comporta como un LFO, el efecto es el de una variación periódica de la altura, lo que se conoce como vibrato, por su relación con los instrumentos acústicos. Pero, el ingeniero y músico John Chowning descubrió que si el modulador oscilaba

a frecuencia de audio ya no se producía una modificación de la frecuencia sino del espectro.

En el caso de modular dos sinusoides entre sí, se producen una serie de bandas laterales alrededor de la portadora. Estas bandas aparecen equidistantes entre sí, a distancia de múltiplos enteros de la frecuencia de la moduladora.

$$f_c + -kf_m$$

Donde f_c es la frecuencia de la portadora (carrier), f_m la frecuencia de la moduladora, y k es un índice entero que varía de 0 a infinito.

El ancho de banda del espectro resultante se puede determinar a partir del índice de modulación definida por la relación entre la cantidad de desviación en frecuencia D (lo que equivale a la profundidad de modulación) y la frecuencia de la moduladora f_m .

Si bien sabemos que el espectro resultante de la FM es infinito, existen solo algunas frecuencias con amplitud significativa. El número de estas bandas es aproximadamente $I + 2$.

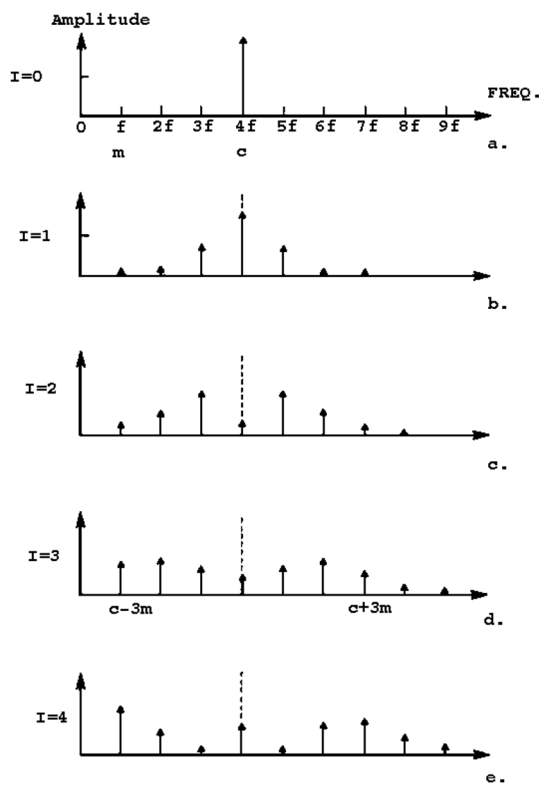


Figura 12: espectro de la FM a medida que crece el índice de modulación I

Las amplitudes de las bandas laterales $J_n(I)$ pueden calcularse, una función de Bessel de primera clase de orden n , evaluada en función del índice de modulación I .

Al ser infinitas, hay ciertas bandas que se reflejan a los extremos del espectro o Hz y $r/2$ (frecuencia de Nyquist). La amplitud de dichas bandas cambian de signo al reflejarse.

Un parámetro muy útil al manejar una FM es lo que suele llamarse grado de armonicidad, determinado por la razón entre la frecuencia de la moduladora y de la portadora.

$$H = f_m / f_c$$

Relaciones enteras devuelven espectros armónicos, en cambio razones más complejas devienen en espectros inarmónicos.

3.1. Instrumento de FM simple

En la Figura 14 vemos una FM simple, donde los parámetros que puede definir el usuario son: la duración total del sonido, la frecuencia de la portadora, el grado de armonicidad y el índice de modulación, y la envolvente dinámica de la onda resultante.

Sabemos que el índice de modulación determina la complejidad del espectro. Si variamos el espectro en función del tiempo logramos una evolución espectral más cercana a lo que sucede en los sonidos instrumentales o de la naturaleza. El objeto envelope permite editar muy fácilmente dicha envolvente a partir de nodos creados con el mouse.

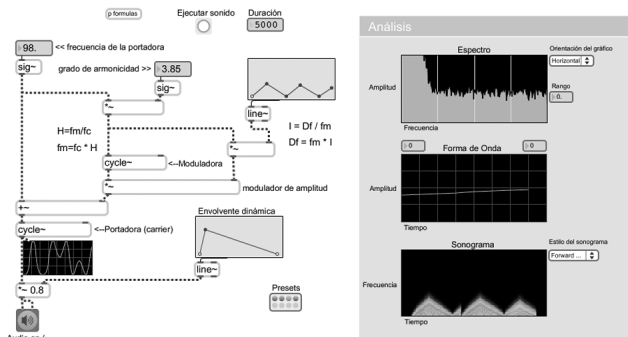


Figura 13

4. Síntesis por modelado físico

La síntesis por modelado físico se vale de modelos matemáticos para simular fuentes reales, en especial instrumentos musicales. Las ecuaciones utilizadas representan leyes físicas que describen, por ejemplo, la forma de producción del sonido, las dimensiones y los materiales del instrumentos, entre otros.

Los modelos de síntesis fueron mejorando, desde los primeros de Hiller y Ruiz, pasando por el algoritmo de Karplus-Strong, y los posteriores desarrollos de Julius O. Smith con la digital waveguide synthesis.

4.1. Algoritmo de Karplus-Strong en Max MSP

Kevin Karplus y Alex Strong desarrollaron, a finales de los años 70, un modelo de síntesis para emular el sonido de las cuerdas pulsadas. El algoritmo está conformado con una línea de retardo retroalimentada, y un filtro pasa bajos dentro del loop de realimentación.

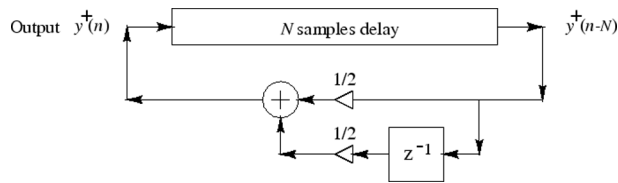


Figura 14: diagrama de flujo del algoritmo de Karplus-Strong. (Imagen obtenida en: https://ccrma.stanford.edu/~jos/SimpleStrings/Karplus_Strong_Algorithm.html)

El filtro pasa bajos cumple una doble función. Por un lado, atenúa las frecuencias más altas del sonido, cada vez que la señal vuelve a dar una vuelta. Esto es lo que en la realidad sucede en los instrumentos de cuerda pulsados, que durante el ataque contienen un sonido con una banda de ruido en el ataque, que luego va perdiendo parciales a medida que transcurre el tiempo. Por otro lado, el filtro afecta levemente la línea de retardo y, como consecuencia la percepción de la altura.

La ventaja de este modelo deviene tanto de su simpleza como de su efectividad, ya que es un algoritmo sumamente eficiente en ahorro de recursos computacionales.

Se genera una onda muy corta, con una duración de N muestras. Si bien, el algoritmo original indica utilizar un generador de ruido blando, suele utilizarse cualquier señal con un espectro rico en parciales. En el caso implementado, la fuente de sonido está conformada por un generador de ruido blanco rand~ multiplicado por una senoide, lo que genera una banda de ruido limitada, centrada sobre la frecuencia de la senoide.

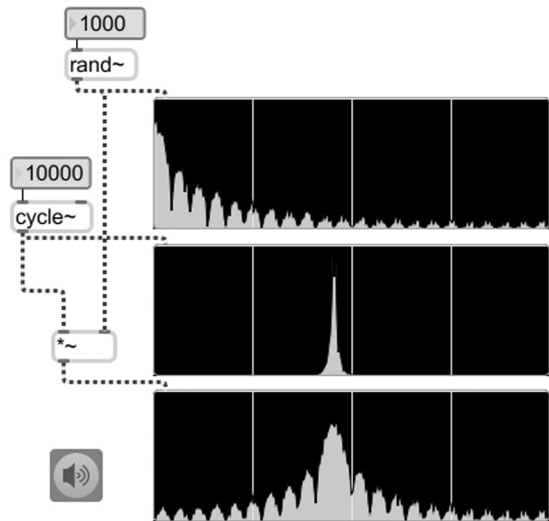


Figura 15: banda de ruido limitada y centrada sobre una frecuencia

La onda de excitación es enviada a una línea de retardo, controlada por el par de objetos tapin~ y tapout~, con un tiempo de delay igual a N.

La salida de la línea de retardo alimenta luego a un filtro pasa bajos de primer orden, con una ganancia menor a 1. El filtro está construido de una forma muy simple, con la suma de la señal que sale de la línea de retardo más una copia de la misma señal, pero retrasada una muestra.

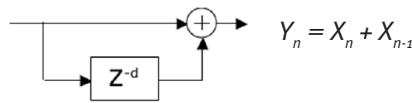


Figura 16: diagrama y ecuación en diferencias del filtro FIR de primer orden

Finalmente la salida del filtro realimenta la línea de retardo, además de dirigirse a la salida.

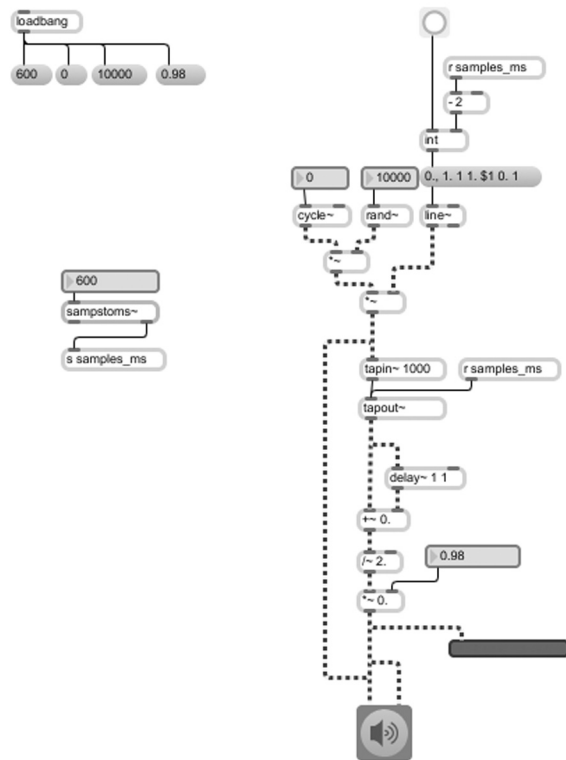


Figura 17: algoritmo de Karplus-Stron. Programación de Pablo Cetta

5. Síntesis granular

La construcción de un evento sonoro complejo a partir de pequeños fragmentos de sonido (granos), de entre 1 y 100 milisegundos, distribuidos en el tiempo se conoce como síntesis granular. La fuente original puede ser un sonido sintético (una senoide, un sonido de FM, etc.), o leído de un sampler (un archivo de sonido leído de un buffer). Básicamente esta técnica consiste en aplicar pequeñas envolventes dinámicas, para crear los granos, y disponerlos en el tiempo de maneras diferentes, de tal

manera que se pueda controlar de manera independiente la altura y la duración del evento sonoro, o bien crear nubes de sonidos, con una distribución aleatoria de los granos.

Esta técnica tiene su origen en la concepción del sonido como un conjunto de partículas dispuestas en el tiempo, conceptos que provienen del paradigma de la física cuántica. La idea de “grano” o “quantum” relacionada con el sonido fue propuesta por primera vez por el físico inglés Dennis Gabor en 1947, pero la primera implementación digital de la síntesis granular fue hecha por Curtis Roads en 1974, en la Universidad de California, y luego desarrollada en 1981 en el MIT.

El compositor griego Iannis Xenakis fue uno de los primeros en introducir el concepto de los granos de sonido en la música, y la utilizó por primera vez en su obra *Analogique A-B*, para orquesta de cuerdas y cinta.

Entre los parámetros que se pueden controlar en un instrumento de síntesis granular aparecen:

- El tamaño del grano
- La forma de la envolvente (gaussiana, quasi gaussiana, trapezoidal, etc.)
- La forma de onda del sonido dentro del grano
- La frecuencia de onda dentro del grano
- Dispersión en el tiempo de los granos

Existen varios modelos de implementación de esta técnica, diferenciados esencialmente por la organización que hacen de los granos.

5.1. Síntesis granular sincrónica por altura

Esta técnica se constituye desde el análisis y resíntesis de un sonido. Del análisis de altura se deduce la frecuencia fundamental del sonido, y de allí su período, que será tomado como unidad o grano. A estas unidades se les realiza un análisis espectral, del que se extrae la respuesta a impulso de dicho espectro. Estas respuestas a impulso controlan los parámetros de un banco de filtros en la resíntesis. La señal de salida resulta de un tren de pulsos con un período igual al del análisis, que excitan el banco de filtros de respuesta a impulso finita (FIR), que actúan como resonadores.

5.2. Síntesis granular cuasisincrónica

Se trata de una o más cadenas de granos puestos en sucesión con diferencias de tiempo variables entre cada uno de los granos. Una envolvente dinámica controla la forma de los granos, que se suceden uno a otro con intervalos de tiempo levemente diferentes.

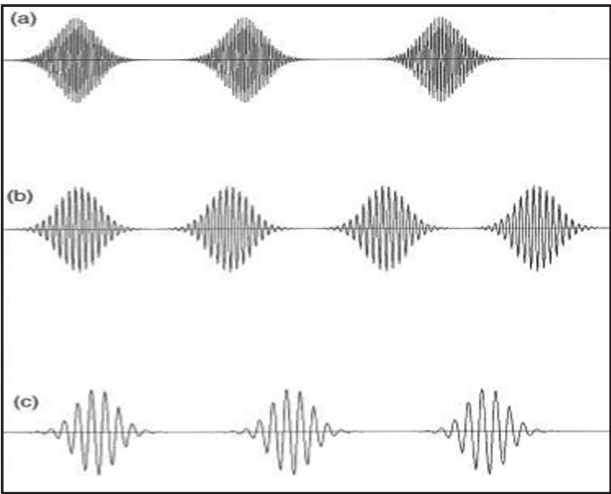


Figura 18: esquema de tres hileras de grains en síntesis granular cuasi asincrónica. La posición en el eje vertical indica la frecuencia del grano. El tiempo entre el inicio de los granos es aleatorio (imagen tomada de Computer music tutorial. Curtis Roads).

5.3. Síntesis granular asincrónica

En la síntesis granular asincrónica los granos son distribuidos de manera estadística en los ejes de la frecuencia y el tiempo. De esta forma pueden crearse “nubes” de granos, en las cuales pueden definirse sus rasgos generales: el comienzo de la nube, la duración de los granos, la densidad (cantidad de granos por segundo), la distribución general de los granos en el registro, la envolvente de amplitud de la nube, la forma de onda dentro de los granos y la distribución espacial de los granos.

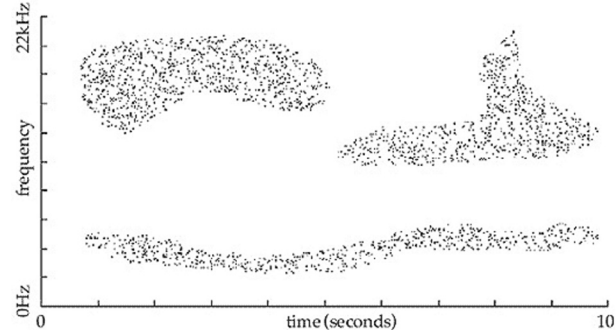


Figura 19: partitura donde se definen “nubes” de granos en función del tiempo y la frecuencia

5.4. Granulación de archivos de sonido grabados

En este caso la envolvente dinámica se aplica a pequeños fragmentos de un archivo de sonido grabado, leídos de un buffer. El control de la lectura de las muestras dentro del *buffer* hace posible distribuciones aleatorias de los fragmentos, reordenamientos en el tiempo, o la mezcla de fragmentos de diferentes archivos, si se utiliza más de un *buffer*.

La flexibilidad de este método hace posible manipular de forma independiente la frecuencia de lectura (*pitch shift*) y la duración del archivo de sonido (*time-stretch*). El ejemplo que sigue es un ejemplo de esta técnica de síntesis granular.

5.5. Un instrumento simple para generación de granos de sonido

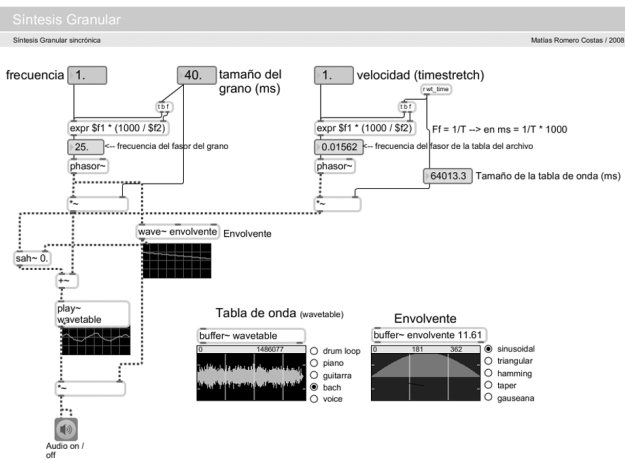


Figura 20

El oscilador objeto `play~ wavetable` es el objeto vinculado a la tabla de onda, del objeto `buffer~` con el mismo nombre, que contiene el archivo de audio a ser procesado. El objeto `wave~`, por su parte, es el oscilador que lee el buffer con la envolvente dinámica que define cada grano.

Este instrumento tiene la capacidad de procesar de manera independiente la frecuencia y la velocidad del archivo de sonido, lo que se conoce respectivamente como *pitch shift* y *time stretch*.

Esto es posible porque existen dos lectores de posición que trabajan en paralelo, mientras que el fasor de la izquierda es el encargado de leer internamente cada uno de los granos, el de la derecha es el que indica la posición general de lectura de la tabla de onda completa. Una vez que el primer fasor finaliza de recorrer un grano, salta a una nueva posición dentro de la tabla. El objeto `sah~` (*sample and hold*) almacena los valores de posición que recibe por su primera entrada, pero solo devuelve un nuevo valor cuando la amplitud de la envolvente que ingresa por su segunda entrada pasa por, o, repite su estado anterior.

6. Síntesis cruzada

La síntesis cruzada consiste en multiplicar los valores de amplitud de los componentes del espectro de una señal por los de otra diferentes. Para decirlo de otra manera, se aplica la envolvente espectral de un sonido al espectro de otro. A diferencia de las técnicas anteriores, donde las operaciones se realizan en el dominio del tiempo, es decir su forma de onda, en la síntesis cruzada, se realiza en el dominio del tiempo, su espectro.

Por convención en las ecuaciones, se utilizan las letras minúsculas para identificar la forma de onda y la mayúscula para su espectro. Por ejemplo:

DFTx(n) = X(k)

Quiere decir que la transformada discreta de Fourier de una onda $x(n)$ nos devuelve su espectro $X(k)$.

Sabemos que si multiplicamos dos formas de onda, el espectro resultante no es equivalente a la multiplicación de ambos espectros, sino a la convolución de los mismos. De la misma manera, la multiplicación de dos espectros es equivalente a la convolución de sus formas de onda (la operación de convolución es conmutativa, al igual que la multiplicación).

h(n).x(n) = H(k)*X(k)
y
H(k).X(k) = DFT(h(n)*x(n))

La convolución lineal o aperiódica se define por la siguiente relación:

y(n) = Σ h(m)x(n - m)
m=0

La convolución se utiliza, muy comúnmente, en la creación de reverberación natural, donde se mezcla un sonido “seco”, de un instrumento por ejemplo, con la respuesta a impulso grabada en un recinto real en particular. Luego de convolucionar ambas señales, el sonido original adquiere las características acústicas de reverberación de dicho lugar.

La forma más común de implementar la convolución en la síntesis cruzada es a través de la FFT, lo que se conoce como convolución rápida. Para ello se siguen una serie de pasos desde los sonidos originales a ser “cruzados” y el resultado final de la síntesis.

- 1. Se realiza la FFT de cada uno de los dos sonidos para pasar de la forma de onda a espectro de la señal.
- 2. Se multiplican los valores de amplitud de ambos espectros.
- 3. Del nuevo espectro obtenido de la operación anterior, se realiza la IFFT (transformada inversa) para obtener la forma de onda del sonido resultante.

6.1. Aplicación en Max MSP

A la izquierda observamos al implementación de la síntesis cruzada de dos archivos de audio leídos y ejecutados por los objetos `sfplay~`. Ambas señales son enviadas al objeto `pfft~`, encargado de la convolución. Los argumentos del objeto se corresponden, respectivamente, con

el tamaño de la ventana de análisis (FFT size), especificado en muestras, y el factor de solapamiento de la ventana de análisis y resíntesis.

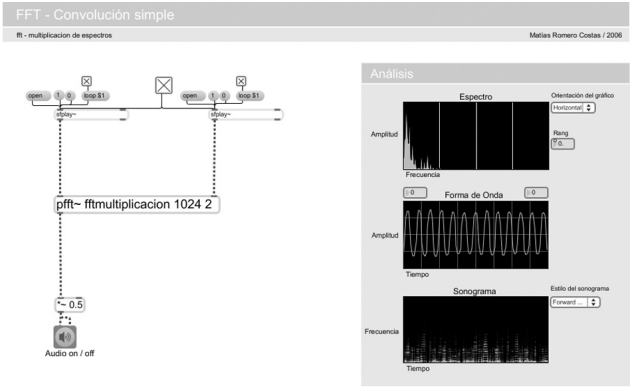


Figura 21

La siguiente figura corresponde a la programación que se encuentra dentro del subpatch `pfft~ fftmultiplicacion`. Los objetos `fftin~` realizan la FFT y devuelven los valores reales (a) e imaginarios (b). Para obtener los valores de amplitud se utiliza el objeto `cartopol~` (que convierte valores expresados en coordenadas cartesianas a polares), a partir de la siguiente ecuación:

$$A = \sqrt{a^2 + b^2}$$

Los valores de amplitud obtenidos son multiplicados por los valores reales e imaginarios de la primera señal, y, finalmente transformados nuevamente al dominio del tiempo a través de la IFFT (transformada inversa).

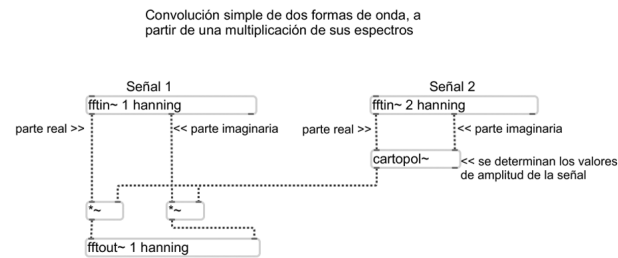


Figura 22: subpatch `pfft~ fftmultiplicacion`

7. Operaciones con líneas de retardo

El valor de salida $y(n)$ es igual al valor de la entrada $x(n)$ retrasado por d muestras. Como el muestreo comienza en el tiempo $t=0$, es decir con la muestra $n=0$, todas las muestras anteriores a $t=0$ no están definidas. Entonces se toma, por lo general, a toda muestra anterior con un valor igual a 0 ($x_{-1}=0$).

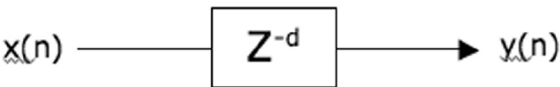


Figura 23. línea de retardo (delay network)

7.1. Delay simple

Un delay simplemente toma una señal de audio de entrada y la retrasa un tiempo determinado (en muestras o milisegundos). Una línea de delay digital se implementa asignando un buffer de valores en memoria, que van a ser leídos un tiempo más tarde. El delay se utiliza para crear ecos, reflexiones en los reverberadores, propagación de ondas, efectos de chorus, flanging, etc.

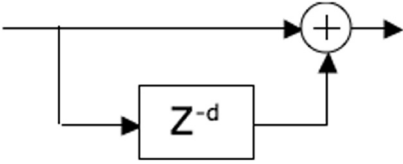


Figura 24: delay simple

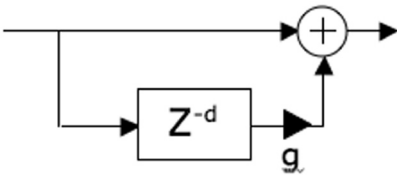


Figura 25: delay con control de ganancia

En Max, los objetos encargados de la línea de retardo son `tapin~` y `tapout~`. Se utilizan siempre juntos, y el primero es el que define el tamaño máximo del buffer y el segundo el tiempo de delay, en milisegundos.

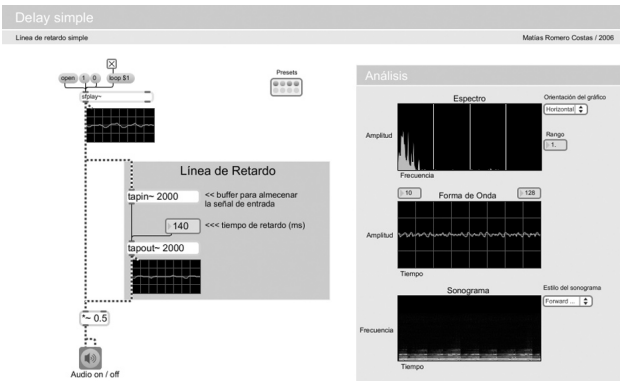


Figura 26

7.2. Delay con realimentación

La realimentación se establece cuando la salida de una línea de retardo vuelve a alimentar su propia entrada. El control de ganancia posterior a la salida del retardo controla el nivel de la realimentación. Con un valor menor a 1 cada nueva entrada en la línea de delay tendrá un valor de amplitud menor (con $g>1$ el sistema se vuelve inestable). En la figura 27 la señal original es sumada a la salida de la línea de retardo; ésta a su vez es nuevamente ingresada en su entrada; la entrada estará formada ahora por la suma de la entrada original y la

salida del retardo, multiplicado por un valor g antes de reingresar a la cadena. Una señal con un valor original de amplitud igual a 1, luego de ser retrasada y multiplicada por g tendrá una amplitud igual a g_1 ; después de ser multiplicada la segunda vez su amplitud, antes de realimentar el delay, será igual a g_2 ; la tercera vez igual a g_3 ; y así sucesivamente. Quiere decir que la señal será multiplicada por g^n , donde n es igual a la cantidad de veces que fue retrasada y multiplicada.

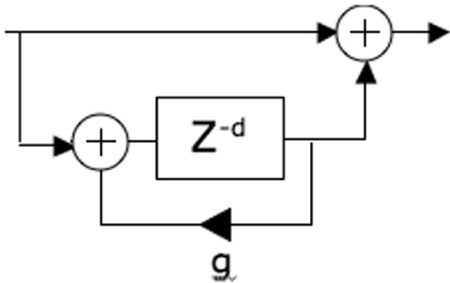


Figura 27. Delay con realimentación

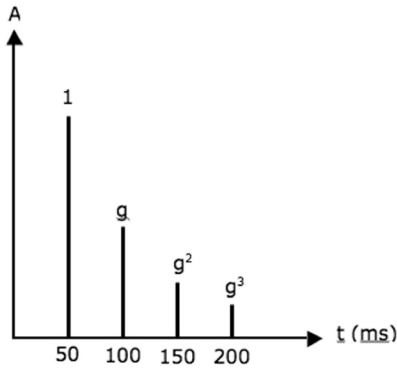


Figura 28. decaimiento de la amplitud de cada una de las repeticiones. El tiempo de retardo es igual a 50 ms.

7.3. Tap-Delay

Se trata de una línea de retardo que permite el acceso en cualquier lugar intermedio de la línea de delay. En el delay anterior la salida es tomada luego de finalizado el retardo total, sin embargo, es posible también acceder a porciones menores del tiempo de delay. A la acción de tomar la salida desde puntos dentro de la línea de delay se las denomina “*tapping*” (*tap* significa llave). En su implementación en software el nombre del delay está relacionado con la cantidad de puntos intermedios (taps) que posea. Así por ejemplo un denominado 3-tap-delay tiene el siguiente esquema:

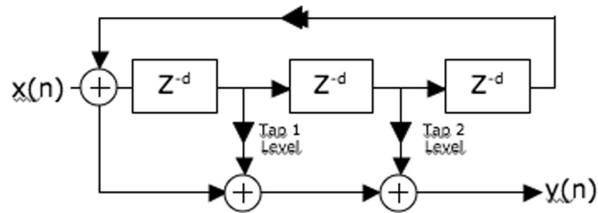


Figura 29: 3 tap-delay

El objeto tapout~ permite implementar múltiples retardos, tantos como argumentos se definan. En este caso un 10 multitap-delay con realimentación de todas las salidas. Cada una de las salidas puede tener un tiempo de retarde diferente, así como su nivel de realimentación.

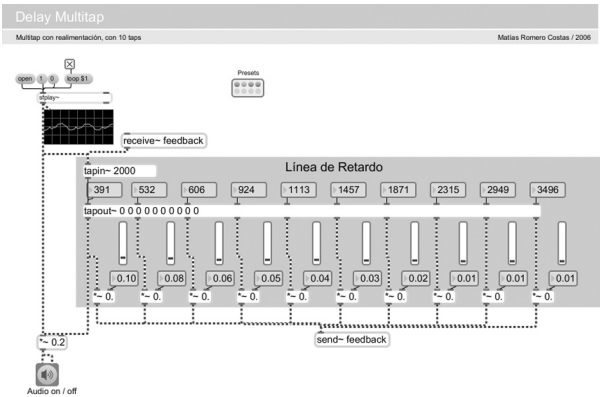


Figura 30

7.4. Filtro comb

Un caso simple de línea de retardo con realimentación son los filtros comb, que toman su nombre por la forma de su respuesta en frecuencia que asemeja a los dientes de un peine (*comb* significa peine).

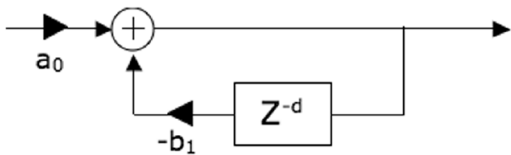


Figura 31: filtro comb con realimentación

Su ecuación en diferencias queda definida por:

$$y(n) = a_0 x(n) - b_1 y(n-d)$$

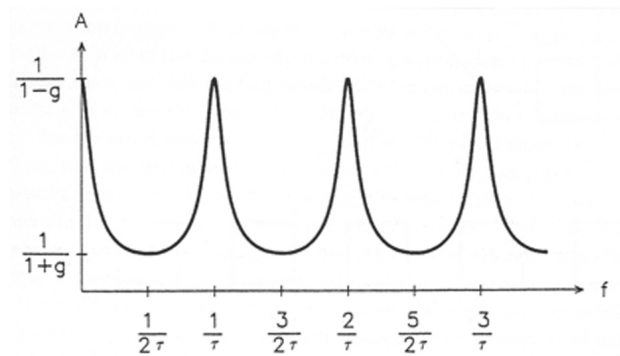


Figura 32: filtro comb con realimentación

Los picos de amplitud de la respuesta en frecuencia están ubicados a una distancia igual a la “frecuencia natural del filtro” que es a la inversa del tiempo de delay:

$$f_0 = 1/\tau$$

La profundidad del mínimo y la altura del máximo dependen de la elección de g , en donde valores más cercanos a 1 significan mayor diferencia entre los extremos. Esto genera que el sonido que entra haga resonar al filtro a su frecuencia natural f_0 , adicionando otro sonido con esa fundamental al original.

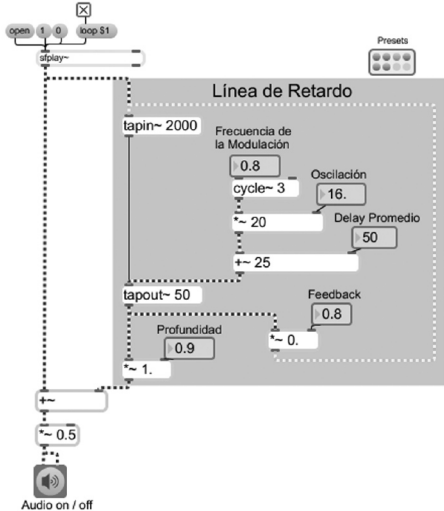


Figura 33: programación de un filtro comb

Si observamos el diagrama, y el algoritmo de Max, podemos notar que el filtro comb es en realidad un delay con realimentación, pero con un valor de retardo muy pequeño, menor a 10 ms. La superposición de la señal junto a copias de sí misma a diferencias temporales tan pequeñas provoca una deformación de la forma de la onda que tiene consecuencias perceptibles al nivel espectral.

7.5. Flanger

El flanger se implementa sumando una señal a una versión retrasada de sí misma con una duración de delay corta y variable en el tiempo. La respuesta en frecuencia, en un instante, es similar a la de un filtro comb, pero donde los picos y ranuras se comprimen y expanden con el tiempo. El espectro de un sonido que pasa por un flanger es enfatizado y atenuado por regiones de frecuencia de forma variable en el tiempo. La duración del retardo es generalmente modulada por un oscilador de baja frecuencia (LFO) y las formas de ondas típicas son la sinusoidal, triangular o exponencial. La profundidad controla entre que valores extenderá el tiempo de retardo.

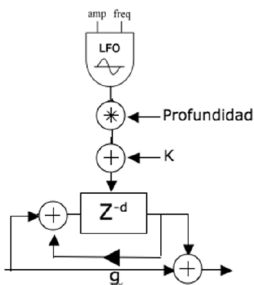


Figura 34

Al escuchar el resultado tan característico de este efecto, y más aun, observando el espectro, podemos entender al flanger como un filtro comb en el cual la frecuencia natural del filtro varía en el tiempo de forma sinusoidal.

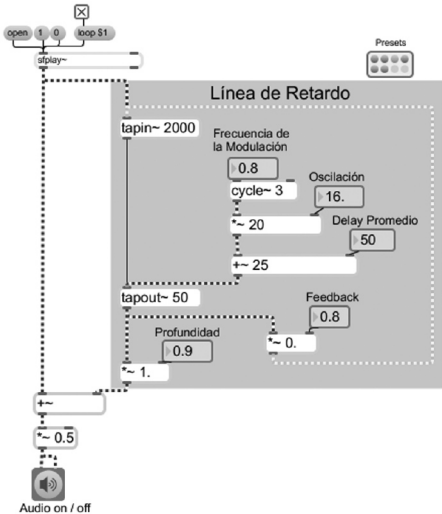


Figura 35: flanger

8. Campanas moduladas

Campanas moduladas es un trabajo realizado en 2009 por los docentes (Pablo Cetta y Matías Romero Costas) y alumnos (Ariel Bottigheimer, Cecilia Chantril, Agustín Genoud, Romina Pow y Leonardo Saltamerenda) de la cátedra de Laboratorio de Sonido III, de la carrera de Artes Multimediales, dentro del Área Transdepartamental de Artes Multimediales del IUNA.

Se trata de una instalación sonora interactiva donde el público se encuentra con unos instrumentos musicales colgantes. Un grupo de pequeñas campanas tubulares suspendidas en el espacio, que pueden ser tocadas por los visitantes. Estos sonidos, que en un principio son los propios del objeto, evolucionan lenta y sutilmente en un complejo sonoro musical.



Figura 36: campanas moduladas.
Exposición “La noche de los museos”.
2009

Cada uno de los seis tubos de la campana tiene colocado un sensor de contacto, conectado a un circuito electrónico, y a una placa arduino. Este sistema de censado está conectado a una computadora, y administrado y controlado desde Max-MSP.

La parte sonora del instrumento, también programada en Max-MSP, está conformada por una serie de módulos de generación y procesamiento de sonido, con 36 configuraciones diferentes, que se suceden automáticamente y de forma aleatoria. Las unidades de procesamiento y generación de sonido utilizados son:

- Players de sonido.
- Multitap delay.
- Flanger.
- FM.
- Modulador en anillo.
- Armonizador.
- Banco de filtros comb.

En la figura siguiente se muestra la pantalla principal del programa. En la parte superior se puede monitorear el estado de los valores de los parámetros de cada módulo de sonido. Abajo y a la izquierda se encuentra el control de nivel general, y los niveles relativos del sonido directo del instrumento y de los procesos. A su derecha la matriz de interconexión, la matriz de presets de configuraciones, el acceso a los procesadores, y la monitorización de los seis sensores de los tubos de la campana.

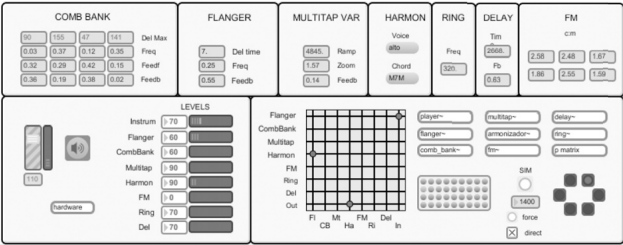


Figura 37: pantalla principal del programa

8.1. Players de sonido

Cada uno de los sensores dispara una copia grabada del mismo sonido del tubo de la campana. Como vemos en el *patch* hay seis objetos *sfplay~* con los sonidos de cada uno de los seis tubos. Cada objeto recibe la orden de ejecutar su sonido cuando el sensor correspondiente es activado.

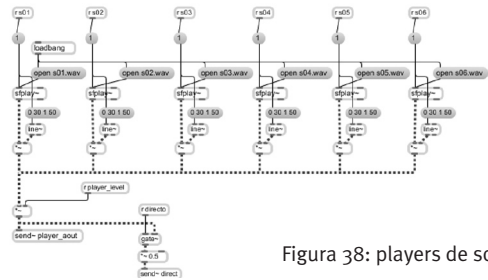


Figura 38: players de sonido

8.2. Multitap con delay variable

Si modificamos el tiempo de delay durante la ejecución, lo que se produce es alteración de la frecuencia. De hecho, existen dos métodos para implementar los transpositores de altura (*pitch shift*): a través de los delays variables, o con un phase vocoder.

El factor de transposición $t[n]$ de la salida de una línea de retardo puede calcularse a partir de:

$$t[n] = y[n] - y[n-1] = 1 - (d[n] - d[n-1])$$

Donde $y[n]$ $d[n]$ es el tiempo de delay (variable). Si $d[n]$ no varía con el tiempo, el factor de transposición $t[n]$ es igual a 1, es decir, no hay cambio de la altura.

Este procesador contiene 10 líneas de retardo variable, con realimentación. Los tiempos de retardo de cada línea son generados aleatoriamente, y pasando de forma gradual del valor actual al próximo con una rampa lineal. Un factor de multiplicación *mt_zoom* actúa a la manera de un “zoom” que permite expandir considerablemente los tiempos de retardo.

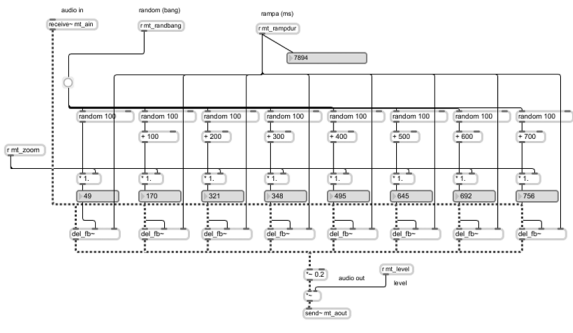


Figura 39

8.3. Flanger

Los valores de delay, frecuencia del oscilador, y nivel de *feedback* del flanger son elegidos aleatoriamente

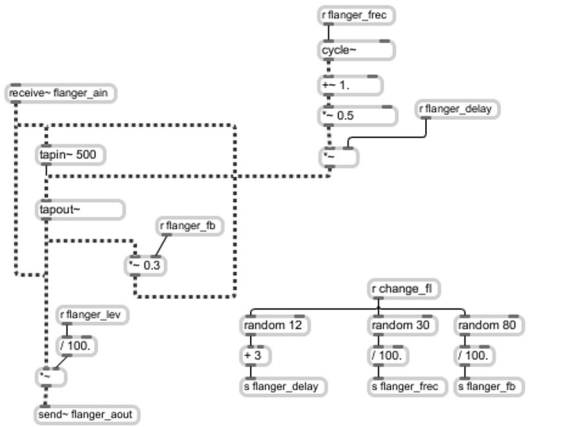


Figura 40

8.4. FM

Aquí la síntesis por FM se utiliza para transformar de manera gradual el sonido original de la campana. Los parámetros iniciales otorgan las características tímbricas que asemejan el sonido sintético al propio del instrumento, pero luego van variando lentamente a espectros disonantes, más cercanos al ruido

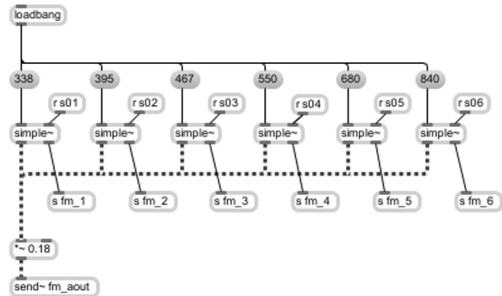


Figura 41

8.5. Modulador en anillo

El parámetro que regula las características tímbricas del sonido a ser procesado es la velocidad de la modulación, es decir la frecuencia a la que oscila la moduladora (objeto cycle~). Los valores posibles van desde 0, sin modulación, hasta 1999 milisegundos. Esto quiere decir que el mismo procesador puede convertirse en una envolvente, un trémolo, o una modulación en anillo, dependiendo de la velocidad a la que se module a la portadora.

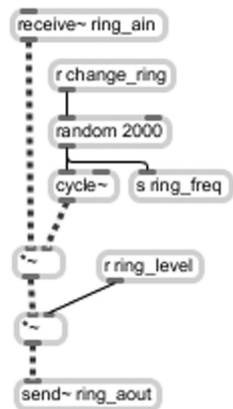


Figura 42

8.6. Armonizador

El armonizador utilizado en este patch es un procesador en el cual cada sonido producido por la campana es armonizado en un acorde de cuatro voces, por lo tanto, posee tres transpositores (figura 43), cada una de estas voces es asignada a una posición en particular del registro y de forma aleatoria, en este caso en particular (bajo, tenor, alto y soprano, (Figura 45). Las posibilidades de armonización de cada una de las voces son de cinco acordes también aleatoriamente (Figura 46).

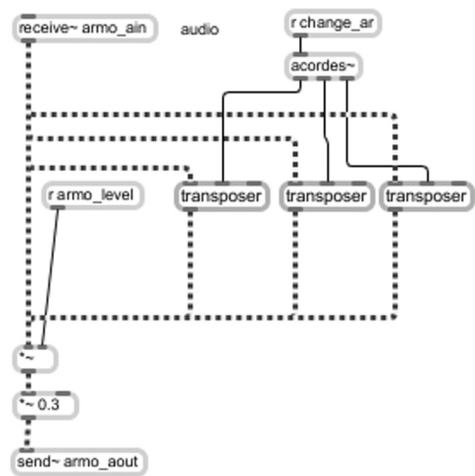


Figura 43: subpatch armonizador

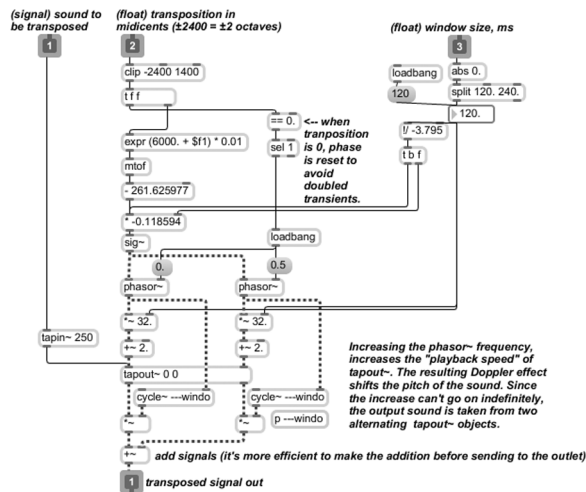


Figura 44: objeto transposer

El objeto transposer es una abstracción que viene con el programa Max-MSP, que permite un rango de transposición de dos octavas hacia arriba y hacia abajo.

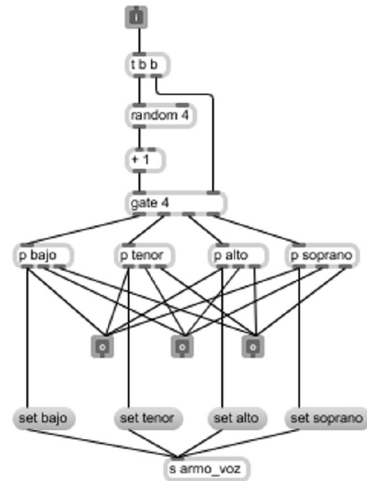


Figura 45: objeto acordes~

El objeto acordes asigna de forma aleatoria el sonido entrante hacia alguna de las cuatro voces (bajo, tenor, alto o soprano).

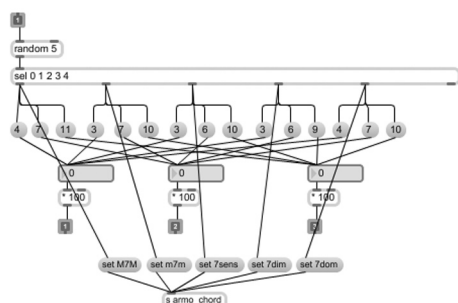


Figura 46: subpatch p bajo

Los subpatch p bajo, p tenor, etc. son los encargados de armonizar los sonidos en cinco acordes diferentes, en este caso en particular pueden ser un acorde mayor con séptima mayor, acorde menor con séptima menor, de séptima de sensible, de séptima dim., y de séptima de dominante. A saber las notas de los acordes están dadas en semitonos y la selección de los acordes es de manera aleatoria.

8.7. Banco de filtros comb

Este procesador está conformado por cuatro filtros comb en paralelo, una característica en particular es que el tiempo de retardo varía en función de un oscilador. Además, podemos controlar el delay máximo, la ganancia del filtro, así como también el nivel del *feed-forward* y del *feedback*, y el nivel general de amplitud.

La ecuación en diferencias implementada en este filtro es:

$$y[n] = ax[n] + bx[n - \text{delay}] + cy[n - \text{delay}]$$

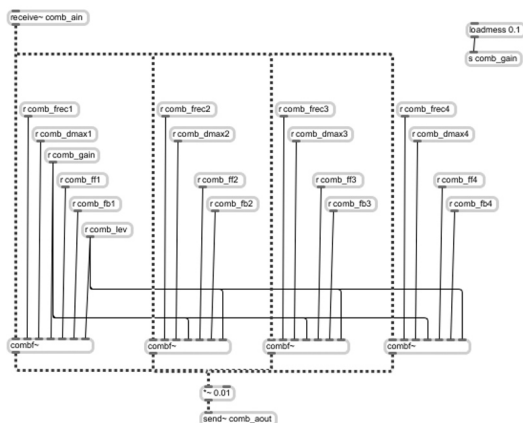


Figura 47: subpatch comb_bank-

8.8. Encadenamiento de las unidades de procesamiento

El programa permite encadenar los módulos de procesamiento entre sí, tanto en serie como en paralelo, mediante una matriz de 8 x 8. Las filas corresponden a las entradas y las columnas a las salidas. El objeto encargado de esto es *matrix~*, que cumple la función de una mezcladora de sonido.

En el gráfico podemos observar en la parte superior las entradas, y en la inferior las salidas. El parámetro *ramp* define un tiempo de crossfade entre las señales (en este caso 2.000 ms), para evitar los clics en el pasaje.

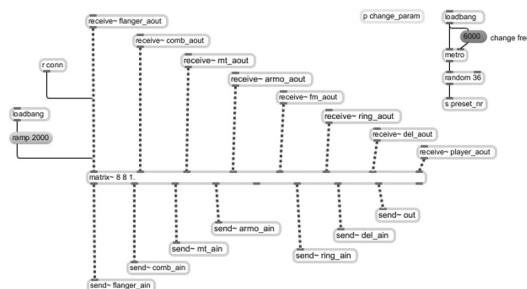


Figura 48

El objeto *matrix~* puede controlarse con un objeto de interfaz gráfica (*matrixcontrol*) que hace más sencillo el manejo de los cambios, al mismo tiempo que facilita la visualización de los estados.

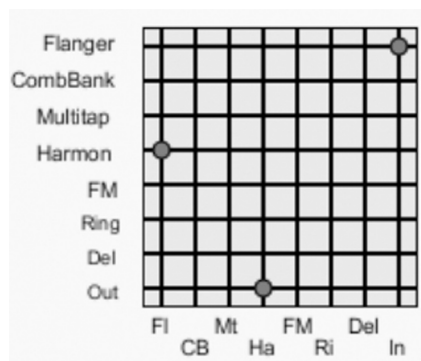


Figura 49

8.9. Automatización de los programas

Es importante remarcar, que al tratarse de una instalación, el sistema funciona de manera totalmente autónoma, sin control alguno por parte de operadores. Una vez puesta en funcionamiento la obra puede permanecer activa un tiempo indefinido, con solo la intervención del público.

Un metrónomo, que se activa al encender la instalación, cambia de programa cada 6 segundos. Cada cambio incluye de que manera se interconectan los procesadores, y los parámetros de configuración de cada uno de ellos.

En la Figura 50 que sigue observamos, como ejemplo, el programa N° 10. Allí el sonido directo (in) pasa a través del flanger (FL) hacia el armonizador (Ha) y, de allí a la salida de audio.

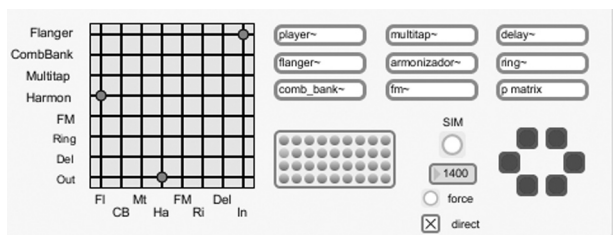


Figura 50: preset de configuración N° 10

Referencias bibliográficas

Chowning, John M. “The Synthesis of complex audio spectra by means of frequency modulation”, en *Journal of the Audio Engineering Society* (California: Stanford, 1973).

Curtis Roads. *The computer music tutorial* (Cambridge, Massachusetts: The MIT Press, 1996).

Di Liscia, Pablo. *Generación y procesamiento de sonido y música a través del programa Csound* (Bernal, Buenos Aires: Editorial de la Universidad Nacional de Quilmes, 2004).

Dodge, Charles y Jerse, Thomas A. *Computer Music* (Estados Unidos: Library of Congress, 1997).

Moore, F. Richard. *Elements of computer music* (Ney Jersey: PTR Prentice Hall Inc., 1990).

Puckette, Miller. *Theory and Techniques of Electronic Music* (San Diego: University of California, 2005).

Smith, Julius O. III. “Physical Modelling Synthesis Update”, en *The Computer Music Journal* (1996: Vol. 20, N° 2, pp. 44-56).

Notas

1. Diagrama de flujo: representación gráfica o esquema simplificado para representar un sistema, sus componentes, la interconexión y la relación entre las unidades generadoras.
2. LFO: Low Frequency Oscillator (Oscilador de baja frecuencia). Es un oscilador que trabaja a frecuencias por debajo del rango audible (20Hz), y que se utiliza como una unidad de control de parámetros en la síntesis de sonido.
3. En las ecuaciones, la operación de convolución se representa con el signo *

(*) www.arduino.cc